S3C6410 Installation Manual for Kernel (linux-2.6.21)

Installation Manual for S3C6410 (Linux)
Copyright © 2008 Samsung Electronics Co, Ltd. All Rights Reserved.

Though every care has been taken to ensure the accuracy of this document, Samsung Electronics Co, Ltd. cannot accept responsibility for any errors or omissions or for any loss occasioned to any person, whether legal or natural, from acting, or refraining from action, as a result of the information contained herein. Information in this document is subject to change at any time without obligation to notify any person of such changes.

Samsung Electronics Co, Ltd. may have patents or patent pending applications, trademarks copyrights or other intellectual property rights covering subject matter in this document. The furnishing of this document does not give the recipient or reader any license to these patents, trademarks copyrights or other intellectual property rights.

No part of this document may be communicated, distributed, reproduced or transmitted in any form or by any means, electronic or mechanical or otherwise, for any purpose, without the prior written permission of Samsung Electronics Co, Ltd.

The document is subject to revision without further notice.

All brand names and product names mentioned in this document are trademarks or registered trademarks of their respective owners.

Contact Address

Samsung Electronics Co., Ltd.
San #24 Nongseo-Ri, Giheung-EUP,
Yongin- City, Gyeonggi-Do, Korea
C.P.O Box #37, Suwon 449-711

Tel: (82)-(031)-209-2378

Email: swhw.hwang@samsung.com
Home Page: http://www.samsungsemi.com

Revision History

Date	Version	Author	Amendment
2008-03-07	1.0	Lee Jongpill	Draft

Contents

1 INTRODUCTION 1

1.1 ARCHITECTURAL OVERVIEW 1

2 GETTING STARTED 1

3 INSTALLING TOOLCHAIN 2

4 BUILDING KERNEL FOR SMDK6410 3

4.1 KERNEL COMPILATION 3

4.2 CHANGING COMPILE OPTION 4

5 DOWNLOADING & WRITING KERNEL IMAGE 7

6 ADDITIONAL INFORMATION 7

6.1 SMDK6410 BOARD LAYOUT 7

Figures

Figure 4-1 Checking Kernel Makefile 3
Figure 6-1 SMDK6410 base board picture 7

1 Introduction

1.1 Architectural Overview

The S3C6410 is a 16/32-bit RISC microprocessor, which is designed to provide a cost-effective, low power, high performance Application Processor solution for mobile phones and general applications. To provide optimized H/W performance for the 2.5G & 3G communication services, the S3C6410 adopts a 64/32-bit internal bus architecture which is composed of the AXI, AHB and APB buses and includes many powerful hardware accelerators for tasks such as motion video processing, audio processing, 2D graphics, display manipulation and scaling. An integrated Multi Standard Codec (MSC) supports encoding and decoding of MPEG4/H.263/H.264 and decoding of VC1. This H/W Encoder/Decoder supports real-time video conferencing and TV out for NTSC and PAL mode.

The S3C6410 has an optimized interface to external memory capable of sustaining the demanding memory bandwidths required in high-end communication services. The memory system has dual external memory ports, DRAM and Flash/ROM/DRAM port. The DRAM port can be configured to support mobile DDR, DDR, mobile SDRAM and SDRAM. The Flash/ROM/DRAM port supports NOR-Flash, NAND-Flash, OneNAND, CF, ROM type external memory and mobile DDR, DDR, mobile SDRAM and SDRAM.

To reduce total system cost and enhance overall functionality, the S3C6410 includes many hardware peripherals such as a Camera Interface, TFT 24-bit true color LCD controller, System Manager (power management & etc.), 4-channel UART, 32-channel DMA, 4-channel Timers, General Purpose I/O Ports, I2S-Bus interface, I2C-BUS interface, USB Host, USB OTG operating at high speed (480Mbps), SD Host & High Speed Multi-Media Card Interface and PLLs for clock generation.

For Small Form Factor applications, POP(Package on Package) option with MCP is allowed.

The ARM subsystem is based upon the ARM1176JZF-S core and includes separate 16KB Instruction and 16KB data caches and 16KB Instruction and 16KB Data TCM, as well as a full MMU to handle virtual memory management. The ARM1176JZF-S is a single chip MCU which includes support for JAVA acceleration. The ARM1176JZF-S includes a dedicated vector floating point coprocessor allowing efficient implementation of various encryption schemes as well as high quality 3D graphics applications.

The S3C6410 adopts the de-facto standard AMBA bus architecture. These powerful, industry standard features allows the S3C6410 to support many of the industry standard Operating Systems.

By providing a complete set of common system peripherals, the S3C6410 minimizes overall system costs and eliminates the need to configure additional components. The S3C6410 is implemented using an advanced 90nm CMOS process. The low-power, simple, elegant and fully static-design scheme is particularly suitable for cost sensitive and power-sensitive applications.

2 Getting Started

Please log in into Super user mode and add the user.
For egs, to add new user as '**test'** and please follow the command below.

```
[root@localhost root]# adduser test
[root@localhost root]# passwd test
Changing password for user test:
New Password:          --Enter the password for user 'test'. And confirm it by entering once again.
```

Copy it to the working directory */home/test*.

Filename	Description
cross-4.2.2-eabi.tar.bz2	Toolchain for Compilation
s3c-linux-2.6.21.tar.bz2	Kernel
s3c-u-boot-1.1.6.tar.bz2	Bootloader
Qt1.7_2nd.cramfs	Qtopia window Root file System

```
[root@localhost test]# ls
cross-4.2.2-eabi.tar.bz2
s3c-linux-2.6.21.tar.bz2
s3c-u-boot-1.1.6.tar.bz2
Qt1.7_2nd.cramfs
[root@localhost test]#
```

3 Installing Toolchain

Building the tool chain is not a trivial exercise and for most common situations pre-built tool chains already exists. Unless you need to build your own, or you want to do it anyway to gain a deeper understanding, then simply installing and using a suitable ready-made tool chain is strongly recommended.

Please follow the commands below and Install the tool chain in the directory mentioned below.

```
[root@localhost test]# mkdir –p /usr/local/arm
[root@localhost test]# tar jxvf cross-4.2.2-eabi.tar.bz2
```

The above commands will generate the **"4.2.2-eabi"** folder under the /test/ directory. Move this folder under **"/usr/local/arm/"** directory.

```
[root@localhost test]# mv 4.2.2-eabi /usr/local/arm/
```

```
[root@localhost test]# export PATH=$PATH:/usr/local/arm/4.2.2-eabi/usr/bin
```

The toolchain object files such as arm compilers, loaders etc. will be available in the *'/usr/local/arm/4.2.2-eabi/usr/bin'* directory.

4 Building Kernel for SMDK6410

4.1 Kernel compilation

Kernel source is compressed by the name of *'s3c-linux-2.6.21.tar.bz2'*. Extract this bz2 file by executing the following command. After extracting the kernel tarball file 's3c-linux-2.6.21' directory will be generated. Go to 's3c-linux-2.6.21.tar' directory and check **'Makefile'** and '**.cross_compile**'.

```
[root@localhost test]# tar jxvf s3c-linux-2.6.21.tar .bz2
[root@localhost test]# cd s3c-linux-2.6.21
[root@localhost s3c-linux-2.6.21]# vi Makefile
CROSS_COMPILE = $(shell if [ -f .cross_compile ]; then \
                        cat .cross_compile; \
                        fi)
[root@localhost s3c-linux-2.6.21]# cat .cross_compile
/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-
[root@localhost s3c-linux-2.6.21]#
```

```
# Another way is to have ARCH set in the environment.
# The default ARCH is the host where make is executed.

# CROSS_COMPILE specify the prefix used for all executables used
# during compilation. Only gcc and related bin-utils executables
# are prefixed with $(CROSS_COMPILE).
# CROSS_COMPILE can be set on the command line
# make CROSS_COMPILE=ia64-linux-
# Alternatively CROSS_COMPILE can be set in the environment.
# Default value for CROSS_COMPILE is not to prefix executables
# Note: Some architectures assign CROSS_COMPILE in their arch/*/Makefile

ARCH            := arm
CROSS_COMPILE   := $(shell if [ -f .cross_compile ]; then \
                           cat .cross_compile; \
                           fi)

# Architecture as present in compile.h
UTS_MACHINE := $(ARCH)

                                                    189,0-1       11%
```

Figure - Checking Kernel Makefile

Set the values by executing *'make smdk6410mtd_defconfig'* command. You can load default-configuration-file[1] that is composed with values optimized to target board. In the case of kernel, default-configuration-files are located in '*s3c-linux-2.6.21/arch/arm/configs/*' directory.

```
[root@localhost s3c-linux-2.6.21]#make smdk6410mtd_defconfig
[root@localhost s3c-linux-2.6.21]#make
```

After executing above commands, the Kernel image will be created in *'/tftpboot'* and *'s3c-linux-2.6.21/arch/arm/boot'* directory by the name of '*zImage'*.

[1] smdk6400nfs_defconfig(NFS), smdk6400ramdisk_defconfig(ramdisk)

zImage (kernel Image) will automatically copy to /tftpboot directory after compilation. If it is not copied, copy it manually as shown below.

```
[root@localhost s3c-linux-2.6.21]# cd arch/arm/boot/
[root@localhost boot]# cp zImage /tftpboot/
```

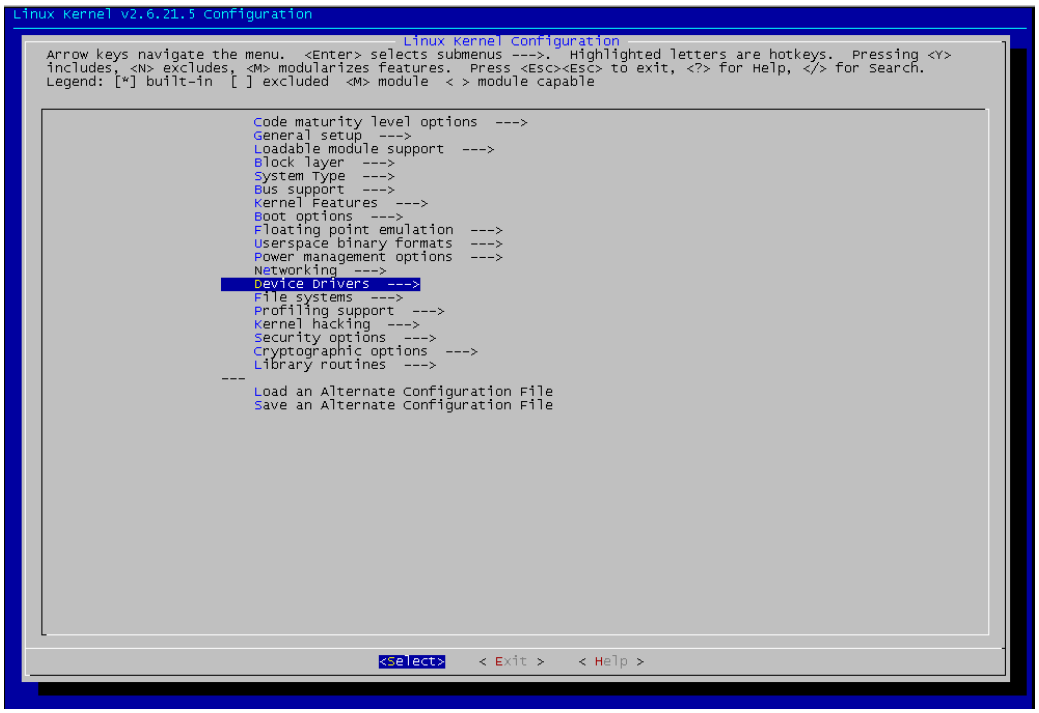
4.2 Changing compile option

If you want to add or remove specific device driver you can choose options in menuconfig screen.

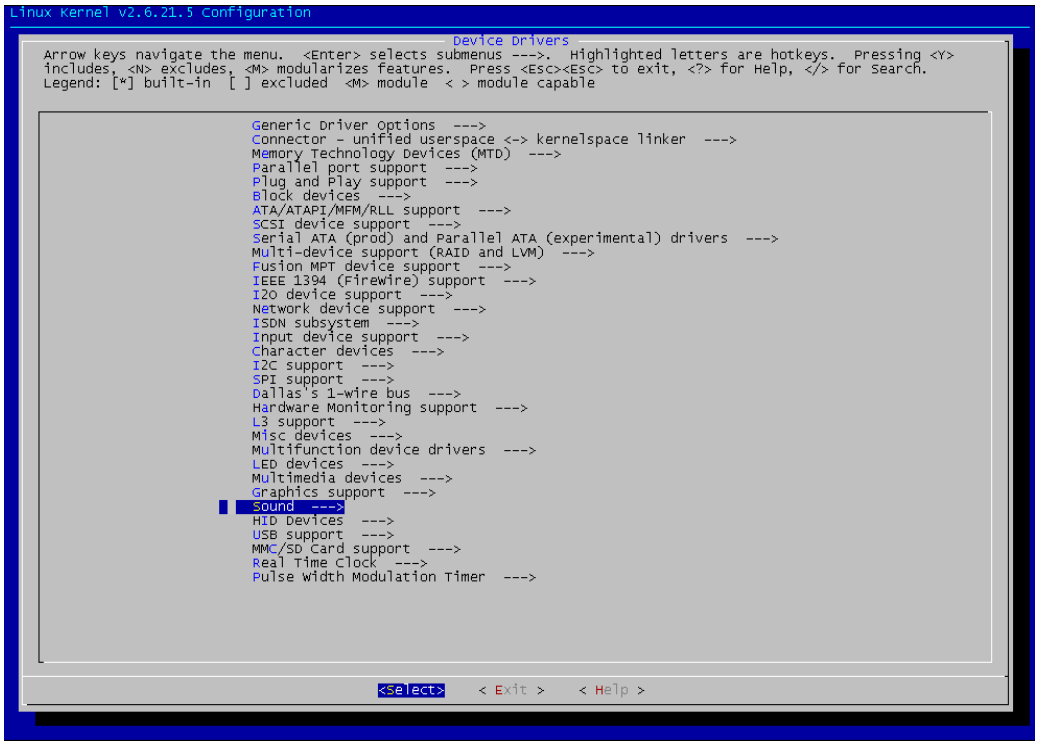
```
[root@localhost s3c-linux-2.6.21]#make smdk6410mtd_defconfig
[root@localhost s3c-linux-2.6.21]#make menuconfig
```

Especially, you don't need to change general options in menuconfig except "Device Drivers" menu. For example, we explains how to include I2S(OSS) device driver.

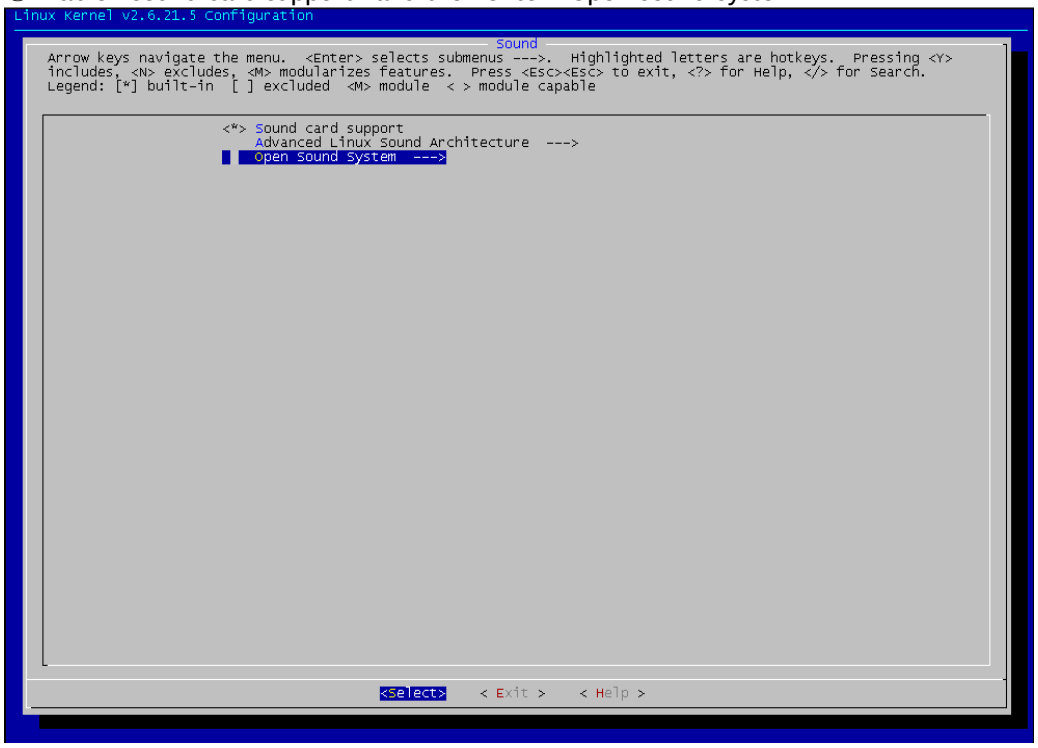
① Select "Device Driver"


Linux Kernel v2.6.21.5 Configuration
Linux Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search.
Legend: [*] built-in [ ] excluded <M> module < > module capable
Code maturity level options --->
General setup --->
Loadable module support --->
Block layer --->
System Type --->
Bus support --->
Kernel Features --->
Boot options --->
Floating point emulation --->
Userspace binary formats --->
Power management options --->
Networking --->
Device Drivers --->
File systems --->
Profiling support --->
Kernel hacking --->
Security options --->
Cryptographic options --->
Library routines --->
Load an Alternate Configuration File
Save an Alternate Configuration File
<Select>
< Exit >
< Help >

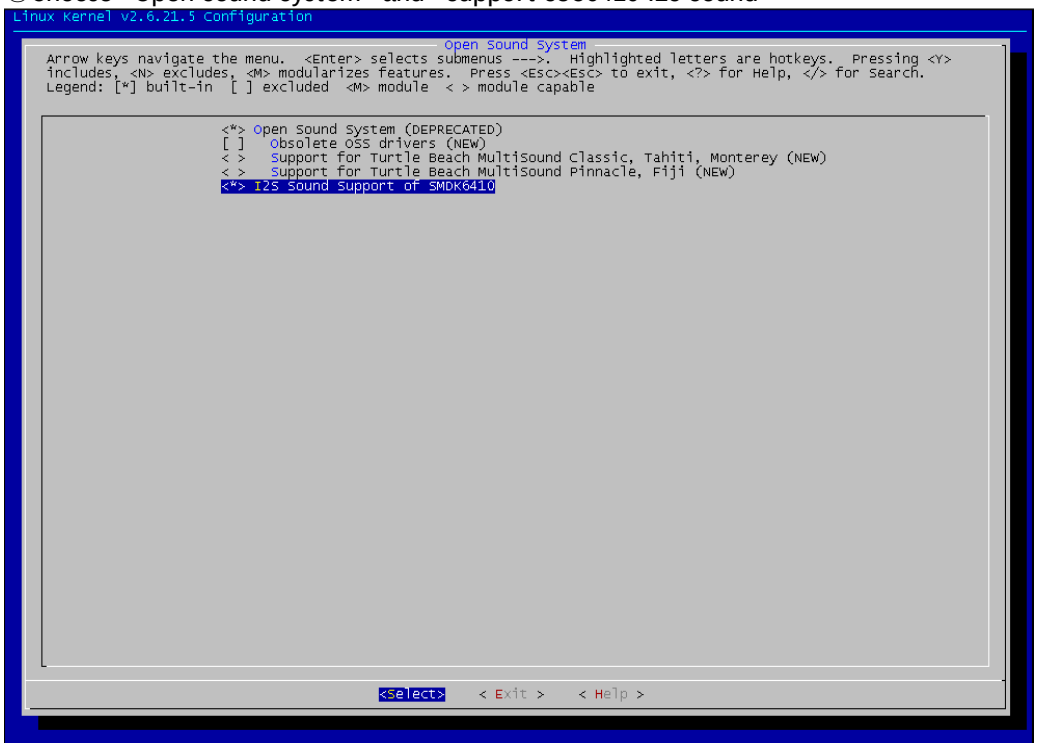
②Select “Sound” menu


Linux Kernel v2.6.21.5 Configuration
Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search.
Legend: [*] built-in [ ] excluded <M> module < > module capable
Generic Driver Options --->
Connector - unified userspace <-> kernelspace linker --->
Memory Technology Devices (MTD) --->
Parallel port support --->
Plug and Play support --->
Block devices --->
ATA/ATAPI/MFM/RLL support --->
SCSI device support --->
Serial ATA (prod) and Parallel ATA (experimental) drivers --->
Multi-device support (RAID and LVM) --->
Fusion MPT device support --->
IEEE 1394 (FireWire) support --->
I2O device support --->
Network device support --->
ISDN subsystem --->
Input device support --->
Character devices --->
I2C support --->
SPI support --->
Dallas's 1-wire bus --->
Hardware Monitoring support --->
L3 support --->
Misc devices --->
Multifunction device drivers --->
LED devices --->
Multimedia devices --->
Graphics support --->
Sound --->
HID Devices --->
USB support --->
MMC/SD Card support --->
Real Time Clock --->
Pulse Width Modulation Timer --->
<Select> < Exit > < Help >

③Enable “Sound card support” and then enter “Open Sound System”


Linux Kernel v2.6.21.5 Configuration
Sound
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search.
Legend: [*] built-in [ ] excluded <M> module < > module capable
<*> Sound card support
Advanced Linux Sound Architecture --->
Open Sound System --->
<Select> < Exit > < Help >

④Choose “Open Sound System” and “Support S3C6410 I2S Sound”


Linux Kernel v2.6.21.5 Configuration
Open Sound System
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search.
Legend: [*] built-in [ ] excluded <M> module < > module capable
<*> Open Sound System (DEPRECATED)
[ ] Obsolete OSS drivers (NEW)
< > Support for Turtle Beach MultiSound Classic, Tahiti, Monterey (NEW)
< > Support for Turtle Beach MultiSound Pinnacle, Fiji (NEW)
<*> I2S Sound Support of SMDK6410
<Select> < Exit > < Help >

⑤Select “<Exit>” in menuconfig or press escape key in keyboard to exit menuconfig. Save changed configuration and then compile.

5 Downloading & Writing kernel image

You can write kernel image into MTD devices. Refer to u-boot installation manual to know how to write kernel.

6 Additional Information

6.1 SMDK6410 board layout

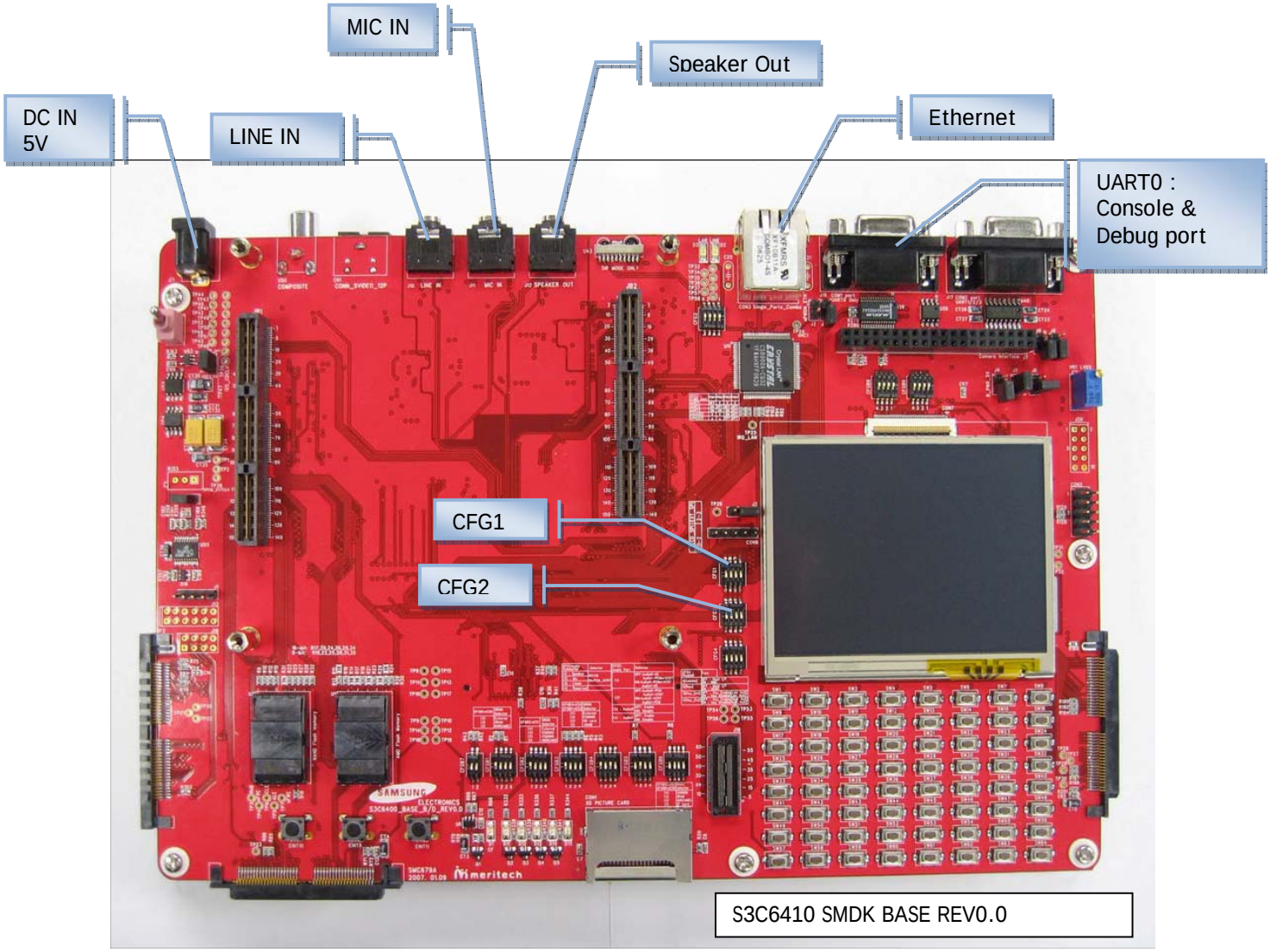


Figure - SMDK6410 base board picture